Class name : SY CSE(IOT)
Rollno : 2007
Urn no : 1022101007
Batch : S1

EXPERIMENT 10

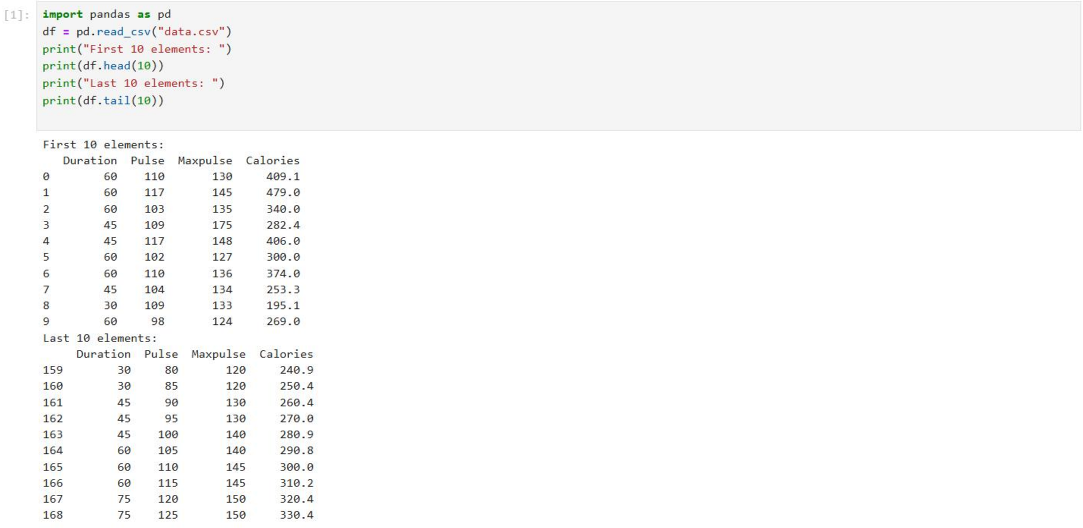
Import any CSV file to Pandas Data Frame and perform the following:

(a) Visualize the first and last 10 records

Code:

```
import pandas as pd
df = pd.read_csv("data.csv")
print("First 10 elements: ")
print(df.head(10))
print("Last 10 elements: ")
print(df.tail(10))
```

OUTPUT:-



```
[1]: import pandas as pd
     df = pd.read_csv("data.csv")
     print("First 10 elements: ")
     print(df.head(10))
     print("Last 10 elements: ")
     print(df.tail(10))
```

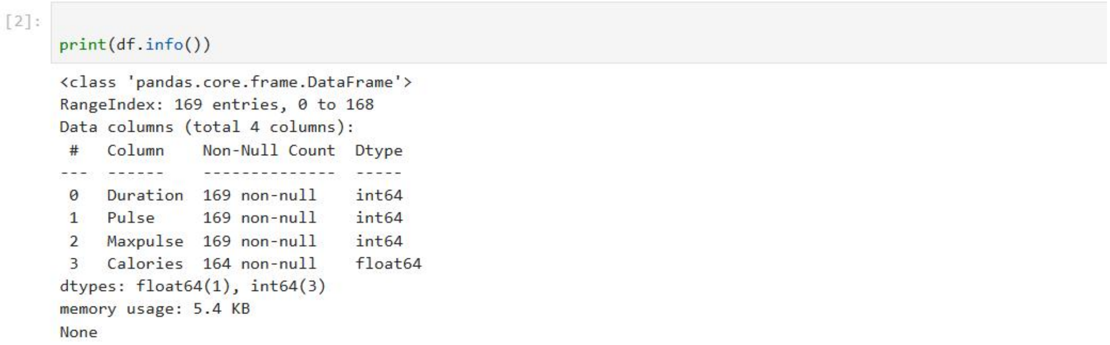
```
First 10 elements: 
   Duration  Pulse  Maxpulse  Calories
0        60    110       130     409.1
1        60    117       145     479.0
2        60    103       135     340.0
3        45    109       175     282.4
4        45    117       148     406.0
5        60    102       127     300.0
6        60    110       136     374.0
7        45    104       134     253.3
8        30    109       133     195.1
9        60     98       124     269.0
Last 10 elements: 
     Duration  Pulse  Maxpulse  Calories
159        30     80       120     240.9
160        30     85       120     250.4
161        45     90       130     260.4
162        45     95       130     270.0
163        45    100       140     280.9
164        60    105       140     290.8
165        60    110       145     300.0
166        60    115       145     310.2
167        75    120       150     320.4
168        75    125       150     330.4
```

(b) Get the shape, index and column details

Code:-

```
print(df.info())
```

OUTPUT:-



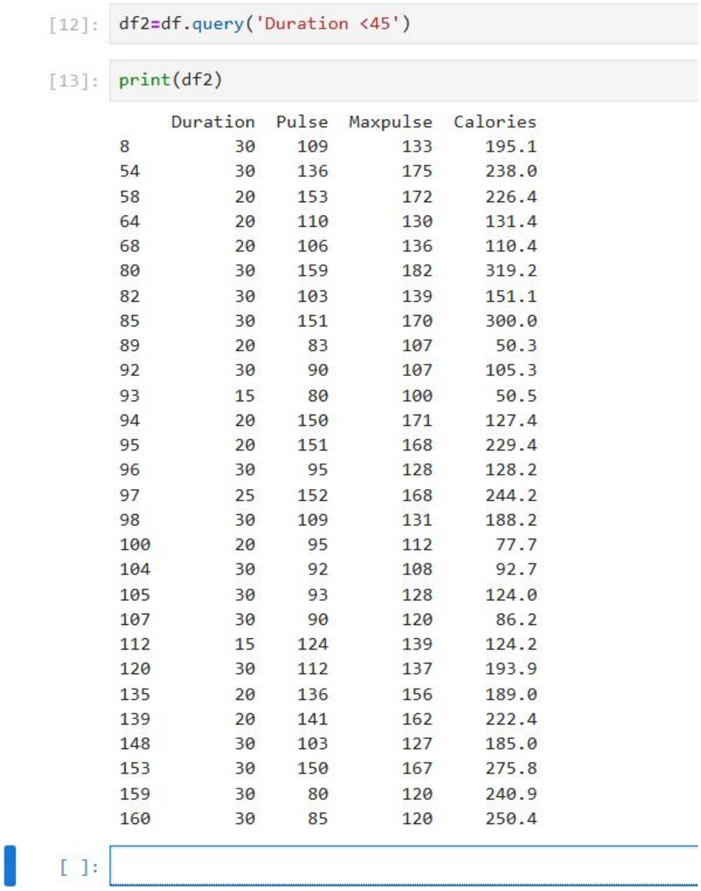
```
[2]: print(df.info())

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 169 entries, 0 to 168
Data columns (total 4 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   Duration  169 non-null    int64  
 1   Pulse     169 non-null    int64  
 2   Maxpulse  169 non-null    int64  
 3   Calories  164 non-null    float64
dtypes: float64(1), int64(3)
memory usage: 5.4 KB
None
```

(c) Select/Delete the records (rows)/columns based on conditions.

Code:-

```
df2=df.query('Duration <45')
print(df2)
```

Output:-



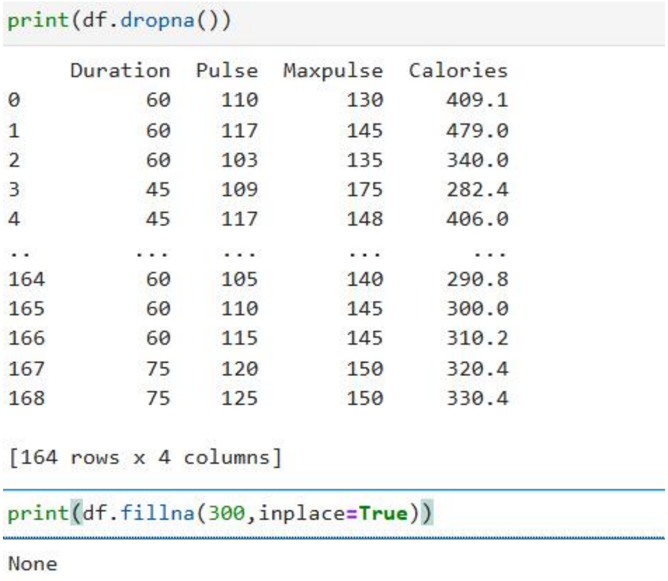
```
[12]: df2=df.query('Duration <45')

[13]: print(df2)

     Duration  Pulse  Maxpulse  Calories
8          30    109       133     195.1
54         30    136       175     238.0
58         20    153       172     226.4
64         20    110       130     131.4
68         20    106       136     110.4
80         30    159       182     319.2
82         30    103       139     151.1
85         30    151       170     300.0
89         20     83       107      50.3
92         30     90       107     105.3
93         15     80       100      50.5
94         20    150       171     127.4
95         20    151       168     229.4
96         30     95       128     128.2
97         25    152       168     244.2
98         30    109       131     188.2
100        20     95       112      77.7
104        30     92       108      92.7
105        30     93       128     124.0
107        30     90       120      86.2
112        15    124       139     124.2
120        30    112       137     193.9
135        20    136       156     189.0
139        20    141       162     222.4
148        30    103       127     185.0
153        30    150       167     275.8
159        30     80       120     240.9
160        30     85       120     250.4

[ ]:
```

(d) Handle missing data by detecting and dropping/ filling missing values.

Code:-

```
print(df.dropna())
print(df.fillna(300,inplace=True))
```

Output:-



```
print(df.dropna())
```

```
     Duration  Pulse  Maxpulse  Calories
0          60    110       130     409.1
1          60    117       145     479.0
2          60    103       135     340.0
3          45    109       175     282.4
4          45    117       148     406.0
..        ...    ...       ...       ...
164        60    105       140     290.8
165        60    110       145     300.0
166        60    115       145     310.2
167        75    120       150     320.4
168        75    125       150     330.4

[164 rows x 4 columns]
```

```
print(df.fillna(300,inplace=True))
```

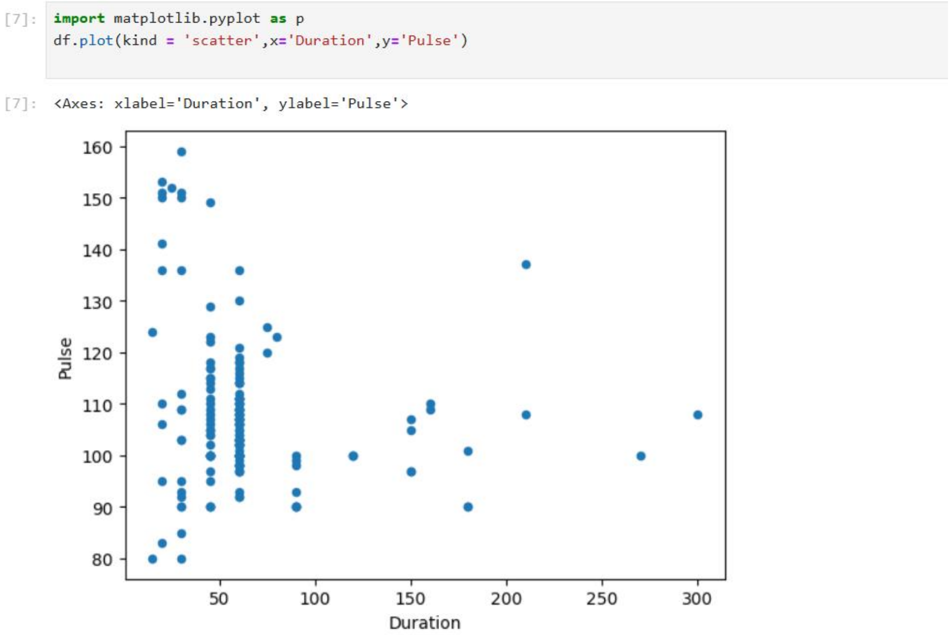
```
None
```

(e) Visualize data using Line Plots, Bar Plots, Histograms, and Scatter Plots.

Code:- # scatter plot

```
import matplotlib.pyplot as p
df.plot(kind = 'scatter',x='Duration',y='Pulse')
```

Output:-



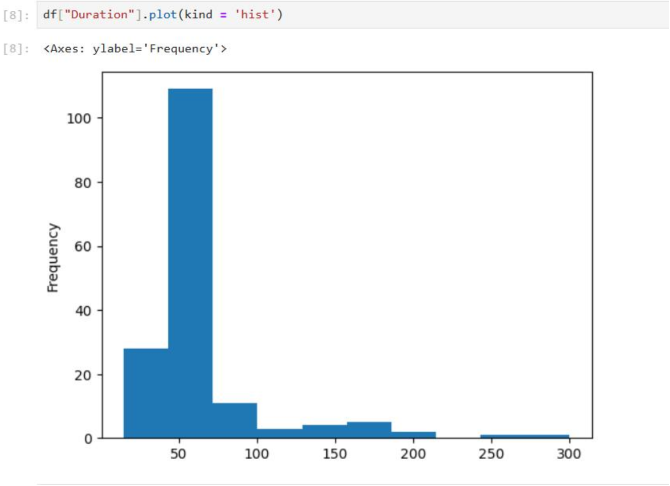
```
[7]: import matplotlib.pyplot as p
     df.plot(kind = 'scatter',x='Duration',y='Pulse')
```

```
[7]: <Axes: xlabel='Duration', ylabel='Pulse'>
```

Code:- #Histogram

df["Duration"].plot(kind = 'hist')

Output:-



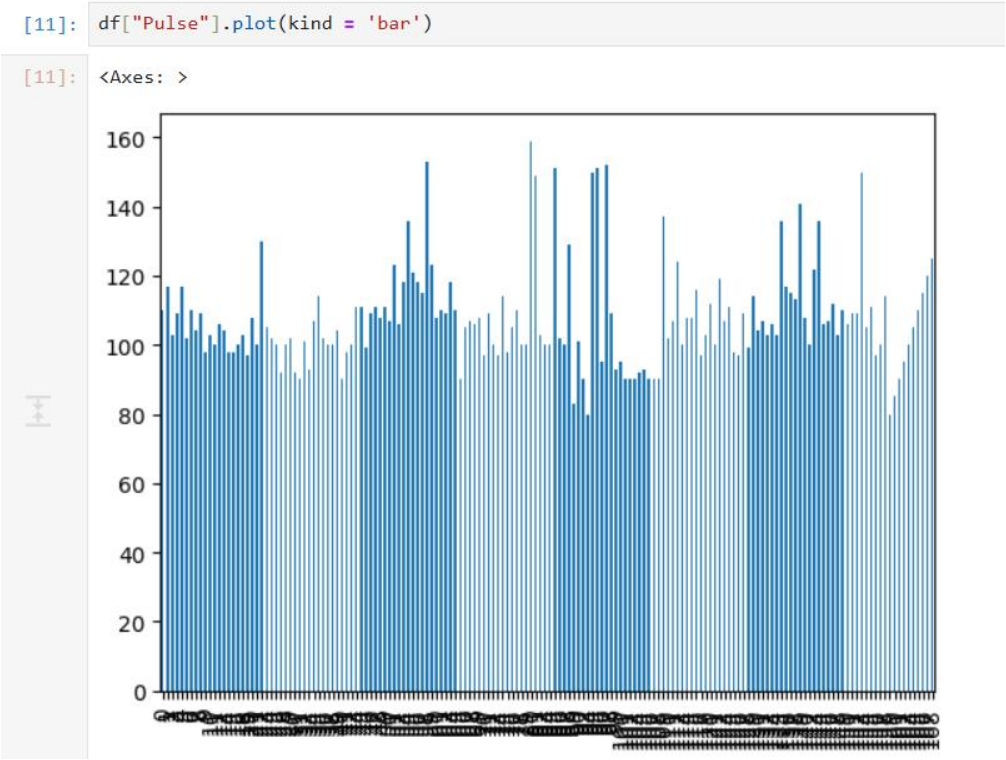
```
[8]: df["Duration"].plot(kind = 'hist')
```

```
[8]: <Axes: ylabel='Frequency'>
```

Code:- #Bar

df["Pulse"].plot(kind = 'bar')

Output:-

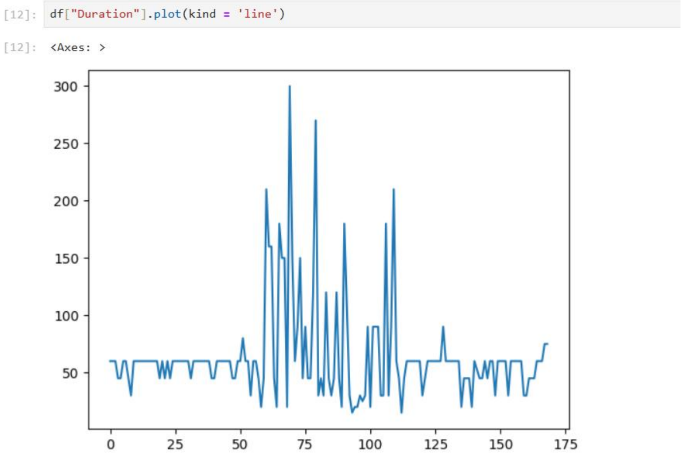


```
[11]: df["Pulse"].plot(kind = 'bar')

[11]: <Axes: >
```

Code:- #line

df["Duration"].plot(kind = 'line')

Output:-



```
[12]: df["Duration"].plot(kind = 'line')

[12]: <Axes: >
```